In the Claims:

Please amend claims 1, 3-5, 7-11, 13-21, 30, 33, and 38-56, please cancel claims 2 and 12, and please add claims 57 and 58, as indicated below.

1. (Currently amended) A system, comprising:

one or more processors;

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement:

one or more applications configured to initiate one or more atomic transactions, wherein each of the one or more atomic transactions comprises requests to access one or more data sources; [[and]]

a transaction manager configured to ~~manage~~ control state changes of the one or more atomic transactions initiated by the one or more applications; wherein for each given atomic transaction, the transaction manager is configured to request permission to change the state of the given atomic transaction[[,]]; and

a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction;

wherein the transaction manager is configured to not change the state of the given atomic transaction without said permission;

wherein the transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction.

2. (Canceled)

3. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is a part of the transaction manager.

4. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager from an administrative entity.

5. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received.

6. (Original) The system as recited in claim 5, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order.

7. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is configured to grant the ~~state transition~~said permission request [[if]]in response to determining that the transaction manager is not paused.

8. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is configured to grant the pause request [[if]]in response to determining that the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request.

9. (Currently amended) The system as recited in claim [[2]]1, wherein the transaction freeze manager is configured to not grant requests if the transaction manager is paused.

10. (Currently amended) A system, comprising a plurality of computer systems coupled by one or more networks, wherein the plurality of computer systems comprise:

one or more processors; and

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers comprising:

one or more applications configured to initiate one or more atomic transactions, wherein each of the one or more atomic transactions comprises requests to access one or more data sources; [[and]]

one or more transaction managers configured to ~~manage~~ control state changes of the one or more atomic transactions initiated by the one or more applications; wherein for each given atomic transaction, the one or more transaction managers are configured to request permission to change the state of the given atomic transaction[[,]]; and

one or more transaction freeze managers configured to pause the transaction manager in response to a pause request by withholding said permission to change the state of the given atomic transaction;

wherein the one or more transaction managers are configured to not change the state of the given atomic transaction without said permission;

wherein the one or more transaction freeze managers are configured to resume the transaction manager in response to a resume request by granting said permission to change the state of the given atomic transaction.

11. (Currently amended) A system, comprising:

one or more processors;

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement:

one or more applications configured to initiate one or more atomic transactions, wherein each of the one or more atomic transactions comprises requests to access one or more data sources; [[and]]

a transaction manager configured to ~~manage~~ control state changes of the one or more atomic transactions initiated by the one or more applications; wherein for each given atomic transaction, the transaction manager is configured to request a read lock on a stored transaction freeze object to change the state of the given atomic transaction[[,]]; and

a transaction freeze manager configured to pause the transaction manager in response to a pause request by withholding said read lock for said stored transaction freeze object;

wherein the transaction manager is configured to not change the state of the given atomic transaction without said read lock; and

wherein the transaction freeze manager is configured to resume the transaction manager in response to a resume request by granting said read lock for said stored transaction freeze object.

12. (Canceled)

13. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is a part of the transaction manager.

14. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is configured to receive requests for write locks on the stored transaction freeze object from an administrative entity to pause the transaction manager.

15. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is configured to queue received lock requests in the order received.

16. (Original) The system as recited in claim 15, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

17. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is configured to grant read locks [[if]]in response to determining the transaction manager is not paused.

18. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is configured to grant a write lock [[if]]in response to determining the transaction manager is not paused and there are no outstanding read lock requests received prior to the write lock request.

19. (Currently amended) The system as recited in claim [[12]]11, wherein the transaction freeze manager is configured to not grant locks [[if]]in response to

determining a write lock on the stored transaction freeze object is currently held by an administrative entity.

20. (Currently amended) A system, comprising a plurality of computer systems coupled by one or more networks, wherein the plurality of computer systems comprise:

one or more processors; and

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers comprising:

one or more applications configured to initiate one or more atomic transactions, wherein each of the one or more atomic transactions comprises requests to access one or more data sources; and

one or more transaction managers configured to ~~manage~~ control state changes of the one or more atomic transactions initiated by the one or more applications; wherein for each given atomic transaction, the one or more transaction managers are configured to request a read lock on a stored transaction freeze object to change the state of the given atomic transaction[[,]]; and

one or more transaction freeze managers configured to pause the transaction manager in response to a pause request by withholding said read lock for said stored transaction freeze object;

wherein the one or more transaction managers are configured to not change the state of the given atomic transaction without said read lock;

wherein the one or more transaction freeze managers are configured to resume the transaction manager in response to a resume request by granting said read lock for said stored transaction freeze object.

21. (Currently amended) A method, comprising:

using one or more computers to perform:

receiving a pause request;

pausing a transaction manager in response to the pause request by withholding permission to change the state of one or more transactions managed by the transaction manager[[.]];

receiving a plurality of resume requests; and

resuming the transaction manager in response to the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager.

22. (Original) The method as recited in claim 21, wherein a transaction freeze manager grants and withholds said permission.

23. (Original) The method as recited in claim 22, wherein the transaction freeze manager is a part of the transaction manager.

24. (Original) The method as recited in claim 22, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager from an administrative entity.

25. (Original) The method as recited in claim 22, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received.

26. (Original) The method as recited in claim 25, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order.

27. (Original) The method as recited in claim 22, wherein the transaction freeze manager is configured to grant a state transition permission request if the transaction manager is not paused.

28. (Original) The method as recited in claim 22, wherein the transaction freeze manager is configured to grant a transaction manager pause request if the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request.

29. (Original) The method as recited in claim 22, wherein the transaction freeze manager is configured to not grant requests if the transaction manager is paused.

30. (Currently amended) A method, comprising:

using one or more computers to perform:

receiving a pause request;

pausing a transaction manager in response to the pause request by withholding read locks on a stored transaction freeze object that identifies a respective atomic transaction[[.]];

receiving a resume request; and

resuming the transaction manager in response to the resume request by granting read locks on the stored transaction freeze object that identifies the respective atomic transaction.

31. (Original) The method as recited in claim 30, wherein a transaction freeze manager grants and withholds the read locks.

32. (Original) The method as recited in claim 31, wherein the transaction freeze manager is a part of the transaction manager.

33. (Currently amended) The method as recited in claim 31, wherein the transaction freeze manager is configured to receive requests for write locks on the stored transaction freeze object to pause the transaction manager from an administrative entity.

34. (Original) The method as recited in claim 31, wherein the transaction freeze manager is configured to queue received lock requests in the order received.

35. (Original) The method as recited in claim 34, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

36. (Original) The method as recited in claim 31, wherein the transaction freeze manager is configured to grant a read lock if the transaction manager is not paused.

37. (Original) The method as recited in claim 31, wherein the transaction freeze manager is configured to grant a write lock if the transaction manager is not paused, and there are no outstanding read lock requests received prior to the write lock request, and there are no outstanding read locks.

38. (Currently amended) The method as recited in claim 31, wherein the transaction freeze manager is configured to not grant locks if a write lock on the stored transaction freeze object is currently held by an administrative entity.

39. (Currently amended) A ~~carrier~~computer readable storage medium ~~comprising~~ storing program instructions, wherein the program instructions are computer-executable to:

receive a pause request;

pause a transaction manager in response to the pause request by withholding permission to change the state of one or more transactions managed by the transaction manager[[.]];

receive a resume request; and

resume the transaction manager in response to the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager.

40. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 39, wherein a transaction freeze manager grants and withholds said permission.

41. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is a part of the transaction manager.

42. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager from an administrative entity.

43. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received.

44. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 43, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order.

45. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is configured to grant a state transition permission request if the transaction manager is not paused.

46. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is configured to grant a transaction manager pause request if the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request.

47. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 40, wherein the transaction freeze manager is configured to not grant requests if the transaction manager is paused.

48. (Currently amended) A ~~carrier~~computer readable storage medium ~~comprising~~ storing program instructions, wherein the program instructions are computer-executable to:

receive a pause request;

pause a transaction manager in response to the pause request by withholding read locks on a stored transaction freeze object that identifies a respective atomic transaction [[.]];

receive a resume request; and

resume the transaction manager in response to the resume request by granting read locks on the stored transaction freeze object that identifies the respective atomic transaction.

49. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 48, wherein a transaction freeze manager grants and withholds the read locks.

50. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is a part of the transaction manager.

51. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is configured to receive requests for write locks on the stored transaction freeze object to pause the transaction manager from an administrative entity.

52. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is configured to queue received lock requests in the order received.

53. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 52, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

54. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is configured to grant a read lock if the transaction manager is not paused.

55. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is configured to grant a write lock if the transaction manager is not paused, and there are no outstanding read lock requests received prior to the write lock request, and there are no outstanding read locks.

56. (Currently amended) The ~~carrier~~computer readable storage medium as recited in claim 49, wherein the transaction freeze manager is configured to not grant locks if a write lock on the stored transaction freeze object is currently held by an administrative entity.

57. (New) The system of claim 1, wherein the transaction manager is configured to, without said permission, perform one or more operations associated with a current state of the given atomic transaction.

58. (New) The system of claim 11, wherein the transaction manager is configured to, without said read lock for said stored transaction freeze object, perform one or more operations associated with a current state of the given atomic transaction.